Security Challenges of IoT-Based Smart Home Appliances

Tuomas Tenkanen, Heli Kallio and Janne Poikolainen

Abstract The Internet of Things, IoT, and the related security challenges are reaching homes in the form of smart appliances. If the appliances are compromised, they can be used in botnet attacks against Internet services and potentially cause harm to people and property through the local network, for example, by heating up too much or allowing unauthorized access. The aim of this study was to see how secure these devices are against remote and network attacks. Several devices were tested with attacks coming from the same Wi-Fi network to gain various levels of control of the devices. Their security against a Man-in-the-Middle attack was also studied to see differences in the susceptibility to connect to another access point. Some devices had a command injection vulnerability and several devices connected to an evil twin. These pose significant risks, but securing the home network and keeping the devices updated protect the devices and secure the system and the smart home.

1 Introduction

The technologies enabling what is known as the Internet of Things, commonly shortened to IoT, have existed for nearly two decades, as has the term (Ashton 2009). However, even today, there exists no commonly accepted definition for the term, but it is usually linked to devices that sense and interact with their environment, are uniquely identifiable, and communicate with other devices (Madakam et al. 2015).

T. Tenkanen (✉) · H. Kallio · J. Poikolainen
Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland
e-mail: tuomas.s.tenkanen@jyu.fi

H. Kallio
e-mail: heli.m.kallio@jyu.fi

J. Poikolainen
e-mail: janne.j.poikolainen@jyu.fi

M. Lehto and P. Neittaanmäki (eds.), *Cyber Security: Power and Technology*,
Intelligent Systems, Control and Automation: Science and Engineering 93,
https://doi.org/10.1007/978-3-319-75307-2_16

Currently, there are approximately 5–6 billion IoT devices connected to the Internet, and the number is rapidly increasing, estimated to reach 20 billion in 2020 (Gartner Inc 2016). Along with industrial systems, building automation and vehicles, all of which are more commonly connected, home appliances are also being networked and are controllable by a mobile phone or other, usually wireless, devices. Besides adding to the comfort of users, this also exposes new privacy issues and attack vectors against home appliances and networks, as well as other network-connected devices (Abomhara and Køien 2014).

The purpose of our research was to find out how secure these devices are, and what kinds of threat they may pose toward the users and toward the networks they are connected to.

2 Background

Many manufacturers of physical devices want to have their products connected, but may not have the knowledge required to build secure devices. These smart appliances will become more and more common, but at home, they will not usually be centrally managed. These facts combined, there will be numerous attack vectors and privacy issues (Black Hat 2016).

2.1 Internet of Things

The term IoT has been evolving over the years, and since the late 90s it has been defined in many different ways (Ashton 2009; ITU internet reports 2016; Mattern and Floerkemeier 2010; Gubbi et al. 2013). The earliest visions were based on combining identifying objects with the Internet to build a networked physical world through the use of RFID technology (Sarma et al. 2000). This vision treated the things more as inanimate objects that could be identified in applications, rather than devices with interaction capabilities. Soon, the vision started to include more advanced functionality, such as things with local processing and communication capabilities. Since then, use of the term IoT has spread to include various fields, including industrial, building and home automation. Today, IoT can be described as an umbrella term for the presence of networked things and objects that are able to interact and cooperate with each other, as well as interact with the physical world in some way and produce data for the different applications, both directly to their owners as well as to various cloud services.

IoT is considered the third information revolution, the first two being the emergence of the Internet in the 90s and social networking in this century. In the Internet of today, nearly all information is originally created by humans, by typing text, taking digital pictures or in some other form of recording information. In the near future, data produced by things will exceed human-produced data and humans will become the minority as both data generators and users (Atzori et al. 2010).

In order to achieve a ubiquitous presence, the things need to overcome many obstacles. The things need to reach a sufficient level of capability, but still need to conform to certain limitations. The key abilities of the devices can be compacted into three groups: cyber-physical, computation, and communication. The cyber-physical capabilities of the devices are dictated by each individual use case. The computational and communication capabilities of the devices are subject to certain constraints in many use cases, so they need be explained further.

Some of the root causes of constraints in the devices' computational and communication capabilities are cost, existing infrastructure, and physical size requirements. For example, if a device cannot be main-powered and has a strict requirement for small size, constraints exist for power usage, which leads to constraints on computation capability and sets restrictions on the communication capabilities as well. Since both computation and communication consume energy, fully featured devices are not always an option. The cost perspective is not always as predominant these days, since the price of fully featured devices has plummeted. A good example of this is the Raspberry Pi Zero, a fully featured computer with a price set to 5 US dollars.

From the communication perspective, the devices can first be divided into two groups: resource-rich devices able to conduct communications using a common Internet stack of protocols and resource-constrained devices that need special protocols for communication. As mentioned before, the constraints of the latter, devices can be caused by limited power, processing capabilities or available memory. The constrained devices can use a special set of protocols to connect to IP-based networks such as 6LoWPAN or use non-IP-based Machine-to-Machine protocols. This gives us a three-tier categorization in the network level for IoT devices, as suggested by Kim et al. (2014).

2.2 Smart Home Appliances

Consumer electronics with communication and physical control abilities have not been in the spotlight of academic IoT research. But in the market, more and more of these devices are surfacing from many leading manufacturers of consumer electronics. The reasoning behind adding smart controls to consumer electronics such as refrigerators and washing machines is often that it would allow better home and energy management (Gubbi et al. 2013), as well as improve convenience , comfort, and safety (Ersue et al. 2015). The functionality of the devices is, in most cases, the ability to remotely control and monitor the appliance, but in many cases the functionality is still pretty basic. Some devices support IoT platforms, such as If-This-Then-That (IFTTT), that enable the users to compose automated functionality for the devices. These platforms in most cases are not at a mature level, and gaps can be found among others in interoperability, since a consensus on standardized communications protocol to enable interfacing with heterogeneous devices has not been reached between the manufacturers and the developing communities (Mineraud et al. 2016).

2.3 Home Networking

The presence of Wi-Fi in homes has grown substantially over the last decade, and together with 4G/LTE Internet access, it is only natural that these technologies have been in the forefront of smart consumer electronics applications. In the design of Wi-Fi-based devices, the assumption is that the collected information is only used by the direct owners of the network (Gubbi et al. 2013). Most of the smart appliances on the market today use Wi-Fi as their primary means of communication. In some products, other forms of communication such as ZigBee are used for communication between devices, but these products are usually sold in sets that include a gateway device.

The management and configuration of home automation networks can be provided in three different ways: by professionals who install the hardware, by third-party service providers, or by the residents themselves (Ersue et al. 2015). Most of the smart home appliances fall into the last category. Most devices do not have user interfaces for setup or control, but the initial setup is typically done with provided applications, which in many cases are smartphone applications. The typical workflow of setting up a device is as follows. The device opens a Wi-Fi access point on which the smartphone is then connected and the credentials for the home Wi-Fi-network are provided for the device. When the setup procedure is complete, the device connects to the home network. After setup, the devices can be controlled through the application or other platforms they support.

The security of the home network is essential for the security of the IoT devices. Wi-Fi is a way to interact with the device, both for the owner and the attacker. A home network is often an easier attack target than a business network. The basic security difference between Wi-Fi networks was noted in a warwalking study in Auckland, New Zealand. In a wardriving or warwalking session, a computer is carried through an area and collects data about the wireless networks it perceives (Kyaw et al. 2016; Eldaw et al. 2013). A few studies have looked into what security protocols are used: more secure ones, such as WPA2 and WPA, unsafe ones, such as WEP, or even a plain open network. In Auckland's Central Business District, 77% of WLANs used secure WPA2, whereas in suburban areas, roughly 60% used it (Kyaw et al. 2016). Another study got similar results in suburban areas: about 60% of networks in a residential area were effectively secured using WPA2 Enterprise, WPA2 Personal, or WPA Enterprise (Kyaw et al. 2016).

Securing a network is not an easy task in business environments, even for professionals and home networks have their own unique weaknesses. Even when communication is secured by WPA2, home networks are less controlled than those of companies. For example, a home network is more prone to infecting malware (Denning et al. 2013). In particular, new, unchecked devices, such as guests' mobile devices, are possible entry points for malware to enter the system (Denning et al. 2013). There are many ways to interfere with the Wi-Fi's functionality or breach its security. The most current attacks that threaten the security are eavesdropping and intercepting the communication, brute force attacks to gain an access point's (AP) password, attacking security protocol's functionality, and misconfiguring the systems (Radack and Kuhn 2012).

2.4 Attack Vectors

Various attack vectors against smart home devices can be identified. The service provided by smart home appliances can often be rather easily denied by overpowering their limited computing power with packet floods. Besides creating annoyance and loss of comfort for the users, in some cases, this might create a risk in the physical world, e.g., allowing unauthorized access to locked spaces. The devices can also be controlled with protocol attacks. Ronen et al. (Black Hat 2016) have demonstrated how to take control of a smart lighting system remotely from several hundreds of meters away by exploiting vulnerabilities in the ZigBee Light Link protocol.

In some cases, the devices can temporarily or permanently be forced into joining other networks, and thus to reveal information regarding the original network to the attacker. This might also allow the attacker to install malicious software in the devices. When the device is allowed to rejoin the original network, the attacker has a persistent entry point to the protected parts of this network. With access to the network, the attacker might perform other actions, e.g., disable services or install malicious software. Of course, the controlled devices could be used to perform Distributed Denial of Service (DDoS) attacks toward other systems, as has been shown recently in attacks against websites (KrebsonSecurity 2016) and DNS providers (Dyn Statement 2016).

Forcing the devices to join another network could begin a Man-in-the-Middle (MitM) attack. In this kind of attack, the attacker places himself or herself between two communicating victims. All communication goes through the attacker's system, and he or she can eavesdrop on it, modify it and prevent messages reaching the other party. In International Telecommunication Union's definition of cybersecurity, three factors should be secured: confidentiality, integrity, and availability (International

Telecommunication Union 2008). This triad is often abbreviated as CIA. The Man-in-the-Middle attack endangers these aims. The data is no longer confidential, as the attacker can record all and has gained access to it. Integrity is also compromised, as all data are going through the attacker's system, which can alter the data. Finally, data's availability is endangered, as the attacker can choose to prevent certain data from reaching the others.

3 Research on Smart Home Appliance Security

Our research was performed in a telecommunications laboratory over a family of smart home appliances that can be controlled with an app on a smartphone over Wi-Fi or through cloud-based services. The devices were purchased off the shelf of a local home appliance shop and connected to the laboratory network per the user manuals of each device. The common connection method was to first allow the device to create a Wi-Fi access point, then to join this network with a mobile phone, and last to instruct the device to join the laboratory network.

The laboratory network was connected to the Internet with a desktop computer acting as router and a Wi-Fi gateway. This setup allowed packet capture on the router and complete control over Wi-Fi traffic. The setup for the laboratory experiments is described in Fig. 1.

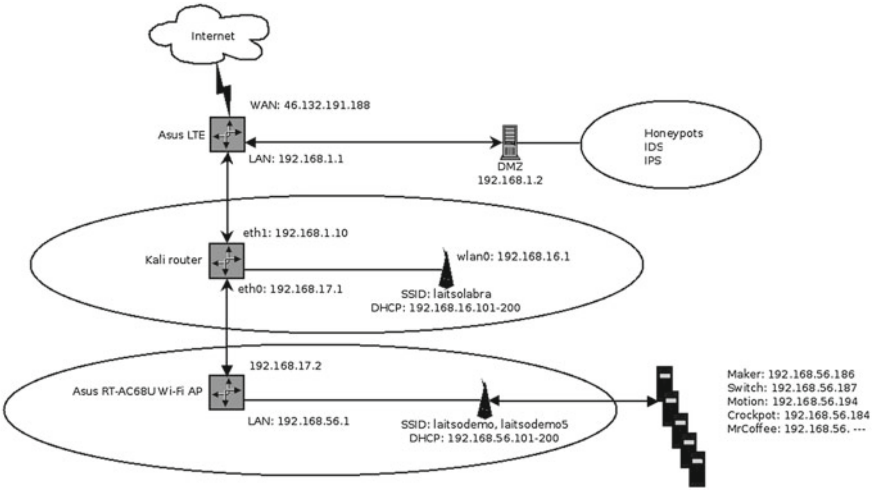

Fig. 1 Laboratory setup

Table 1 Default open ports on the devices

Device	Open TCP ports	Open|filtered UDP ports
A programmable relay	53, 49153	53, 67, 1900
A motion detector	53, 49153, 49154	53, 1900, 49212
An electricity switch	53, 49153	53, 1900
A crockpot	53, 49153, 49155	53, 67, 1900, 18081
A coffee maker	53, 49153, 49154	53, 68, 1900, 4000

3.1 Attacks within the same network

In November 2015, Hart (2015) published a remote attack against this type of device allowing an attacker to gain root shell access to them. An unsanitized input string to a method on the device was used to execute commands to open a telnet daemon. A firmware update fixing the vulnerability was released by the manufacturer in May 2015.

A number of similar devices were acquired for further research: a coffee maker, a crockpot, an electricity switch, a motion detector, and a programmable relay. The devices were installed into a laboratory network as instructed by the manufacturer. The network traffic generated during the install process was recorded at the Wi-Fi gateway and examined later. Port scans of each of the five types of device tested revealed multiple open TCP and UDP ports on the devices, as shown in Table 1.

A closer look at the ports listed and the recorded network traffic reveals the control interface of the devices in the 4915x ports. The interface uses HTTP/SOAP protocol to interact with the smartphone controlling the devices. A TCP SYN flood attack on the control interface allows an attacker to perform a simple but efficient Denial of Service (DoS) against the device. Such an attack requires a restart of both the device and the smartphone app to recover. As easy as it is to perform this attack once within the network, the result is to be expected, as the computers in the appliances are low-specified, and thus not able to compete with a desktop computer using all its power flooding the packages into the network. Interestingly, having recovered from such an attack, the devices alter their control interface port from 49153 to 49155 or vice versa in order to avoid being immediately affected by the same attack should it continue.

As per the UPnP standards, the devices advertise their services and methods into the network they are connected to. This functionality and the unencrypted protocol used allow anyone with access to the same network to perform these methods, e.g., to query firmware versions in the installed devices or, e.g., turn on the heat on a crockpot with properly formatted HTTP/SOAP requests. The firmware versions reported by the devices using these method calls are reported in Table 2.

Table 2 Device firmware versions

Device	Firmware version
A programmable relay	2.00.9898
A motion detector	2.00.1700
An electricity switch	2.00.1700
A crockpot	2.00.6461
A coffee maker	2.00.5607

Hart used a method called "SetSmartDevInfo" with a parameter "SmartDevUrl" to compromise the device (Hart 2015). The presented procedure was not directly replicable, because the laboratory devices did not include a telnetd binary. This was probably due to different firmware versions between ours and Hart's devices. However, a functioning command injection allowing an attacker, e.g., to reboot the devices remotely was found, thus confirming the findings on four of the five devices. The one device not found to be vulnerable to the attack was the programmable relay with firmware version 9898, newer than the one mentioned by Hart, thus further confirming the results found.

Later experimentation with various commands revealed a Linux system being run on the devices. Having access to execute commands and read responses on the network, a web server root directory was discovered, and thus details of the system, directory listings, and executable binaries could be retrieved through the control interface web service. The examination of the binary files revealed the architecture of the devices to be based on MIPS processors and the shell to be busybox. The file listings also included wget, a command-line application for retrieving files from the network.

Busybox is a multi-call binary shell with many optional functions to be compiled into. Apparently, the version included in our devices did not have all the features Hart's version had, and thus did not respond to telnetd commands. The earlier results enabled us to instruct the device to retrieve a pre-compiled complete version of busybox from the network and run it on the device, allowing us to open a telnet port with a root shell. This worked only on the devices with firmware versions before 8643. The bug being absent in later versions was reported by Hart (2015), and these new findings confirm both the vulnerability and the fix.

Several other methods were found to be similarly vulnerable to the unsanitized URL handling. Simple shell scripts for demonstrating the vulnerability and allowing remote root access were created. An example of such a script utilizing a firmware update URL being run and a transcript of this are shown below:

```
./upnp-firmwareversion.sh 192.168.16.194 49153
HTTP/1.0 200 OKCONTENT-LENGTH: 361CONTENT-TYPE: text/xml; charset="utf-
8"DATE: Thu, 03 Mar 2016 13:41:51 GMTEXT:SERVER: Linux/2.6.21, UPnP/1.0,
Portable SDK for UPnP devices/1.6.6X-User-Agent: redsonic<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><s:Body>
<u:GetFirmwareVersionResponse
xmlns:u="urn::service:firmwareupdate:1"><FirmwareVersion>FirmwareVersion:
WW_2.00.1700.PVT|SkuNo:Plugin
Device</FirmwareVersion></u:GetFirmwareVersionResponse></s:Body> </s:Envelope>

./open-telnet-firmware.sh 192.168.16.194 49153
HTTP/1.0 200 OK
CONTENT-LENGTH: 285CONTENT-TYPE: text/xml; charset
="utf-8"
DATE: Thu, 03 Mar 2016 13:42:06 GMT
EXT:SERVER: Linux/2.6.21, UPnP/1.0, Portable SDK for UPnP devices
/1.6.6X-User-Agent:
redsonic

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><s:Body>
<u:UpdateFirmwareResponse
xmlns:u="urn::service:firmwareupdate:1"><status>success</status></u:UpdateFirmware
Response></s:Body> </s:Envelope>

nmap 192.168.16.194
Starting Nmap 7.01 ( https://nmap.org ) at 2016-03-03 13:40 EET
Nmap scan report for 192.168.16.194
Host is up (0.0095s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
53/tcp    open  domain
49153/tcp open  unknown
MAC Address: EC:1A:59:79:A4:39 ()
Nmap done: 1 IP address (1 host up) scanned in 0.72 seconds
```

As can be seen from the transcripts, the first script run calls the method “GetFirmwareVersion” to which the device responds with details and the version of its firmware. The second script then sends an URL within a firmware update method parameter containing shell commands wrapped in a command substitution instruction (Command Substitution 2016). The device does not sanitize the input URL, but rather executes these commands as they are. The port scan afterward reveals an open telnet port that is connected to a password-less root login, allowing complete remote control over the device.

The attack could be spread further. With complete control over the device, other vulnerable devices could easily be found and compromised in the same Wi-Fi. We could also create a service on the compromised device offering the fully featured shell binary of the injected version of busybox. This way, the new compromised devices would get all the tools of busybox fast, even without connection to the Internet and possibly from multiple sources.

Complete control of the device also allows unrestricted access to its file system. This allows any file to be served through the control interface web service, or through

Table 3 /etc/passwd

Lines in /etc/password
root:$1$$CoERg7ynjYLsj2j4glJ34.:0:0:root:/tmp:/bin/sh

Table 4 Partitions using command dmesg|grep^0x0

Address	Description
0x0000.0000–0x0005.0000	"uboot"
0x0005.0000–0x007c.0000	"A—Kernel and Rootfs"
0x0015.0000–0x007c.0000	"A—Rootfs"
0x007c.0000–0x00f3.0000	"B—Kernel and Rootfs"
0x008c.0000–0x00f3.0000	"B—Rootfs"
0x00fe.0000–0x00 ff.0000	"Nvram"
0x00 ff.0000–0x0100.0000	"User_Factory"
0x0004.0000–0x0005.0000	"Factory"
0x00f3.0000–0x00fd.0000	"Manufacturer_settings"
0x0030.0000–0x0004.0000	"Uboot_env"

any other service started for this purpose./etc/passwd is a file containing the usernames and passwords of the system. An electricity switch with firmware version 1700 has its contents shown in Table 3 in the/etc/passwd file.

Running the John the Ripper password cracking tool against the /etc/passwd file reveals the very simple credentials of "root:admin" within seconds, even using just the default options of the password cracking tool. In later firmware versions, a more complex password has been used.

With root access to devices, both mounted and unmounted file systems and partitions could be read and dumped to other computers over the network using ftpput commands and reading directly off the device file. A total of ten file system partitions was discovered in the device. The mounted file systems include /dev/mtdblock2 of type squashfs (rw), /dev/mtdblock8 of type jffs (rw) and ramfs on /tmp. Further examination revealed the rest of the partitions, shown in Table 4.

The partition types further revealed details of the system, showing that the device utilizes a Uboot-system and stores two firmware versions labeled "A" and "B" concurrently. Which one is to be used when booting the device is selected by a setting stored within /dev/mtdblock0. The settings used in normal operation of the device were found to be stored in /dev/mtdblock5, the Nvram partition, with some examples shown in Table 5. Other bits of information not shown include the SSID of the network the device is connected to and the firmware version currently installed, among many others. With some knowledge, the device can rather easily be tricked into thinking that is has the newest firmware available, and thus prevent it from being updated, granting the attacker persistent access to the infected systems.

Table 5 Stored password settings

Strings mtd5|grep-i pass
ppp0_pppoe_passwd=
ppp1_pppoe_passwd=
pppoe_password=
pptp_password=
l2tp_password=
bigpond_password=
wl0_authRadiusPasswd=
ipsec_passthru_enabled = 1
pptp_passthru_enabled = 1
l2tp_passthru_enabled = 1
httpd_password = admin
mradius_password = admin
ddns_password=
login_password = d41d8cd98f00b204e9800998ecf8427e
http_passwd = d41d8cd98f00b204e9800998ecf8427e
ClientPass = eJLIJg7WxAOpilzZ62T95w==

3.2 Man-in-the-Middle Attacks

In a MitM attack, all communication between the victim and other parts of the network goes through the attacker's system, and he or she can eavesdrop on it, modify it and prevent messages from reaching the other party. The MitM attacks have been an effective attack for a long time (Prowell et al. 2010). MitM was listed as the fifth most common technique used in data breaches in Verizon's data breach report of 2011 (Baker et al. 2011). The most common technique was the use of stolen credentials. Next frequent were three types of malware: those capturing data, those sending data from outside the system, and those that install other malware or updates into the targeted system.

Cybersecurity's three factors, abbreviated as CIA, are all endangered by MitM. These factors are confidentiality, integrity, and availability. Prowell, Kraus, and Borkin use the children's game of "telephone" to demonstrate this type of network attack. Children are in a circle or row, and the first child sends a message to the last child by telling the message to the next child, who then tells it to the next until the last child gets the message and says it aloud. In this game, an attacker, a mischievous child, could slip between the others. He or she would hear the message and could then relay the message untouched, modify it to be funnier, or refuse to relay the message to the next child altogether.

As in the telephone game, data's availability is endangered, as the attacker can choose to prevent certain data from reaching others. The data is no longer confidential, as the attacker can record all. Finally, integrity is compromised, as all data are going through the attacker's system, which can alter the data.

However, encryption can secure integrity and confidentiality. If communication is encrypted, the attacker can't read the contents of the messages. Therefore, the actual data stay confidential. Also, it is practically impossible to change encrypted data, so that the recipient would believe it to be a valid message. Dropping data packets is still an option, so availability can still be threatened.

An MitM attack is a broad attack type that can be used on various levels of communication, as Conti, Dragoni, and Lesyk have examined in their survey (Conti et al. 2016). This research focused on Wi-Fi, as all examined appliances used it as their main way of communicating: they connected to home networks wirelessly and even set themselves an open Wi-Fi for initial setup. An MitM attack can be initiated in a Wi-Fi by setting up an access point as an evil twin, and getting the victim to connect to it. Once connected, the data can be accessed and manipulated by the attacker.

An AP posing as another is called an evil twin, or sometimes a rogue or fake AP (Kumar and Paul 2016). In a Wi-Fi network, devices are connected to an access point that is identified by its SSID and BSSID (Kumar and Paul 2016). SSID is the network name and BSSID is the AP's MAC address. These both can be easily spoofed and an attacker can pose as a genuine AP (Lanze et al. 2015; Sheng et al. 2008). Often, the evil AP needs to be physically closer to the victim, or otherwise send stronger signals, as users often connect to the strongest signal indicated by Received Signal Strength Indication (RSSI) (Mustafa and Xu 2014; Song et al. 2010).

3.3 Experiment Design

Seven devices' resilience against an attempted MitM attack were tested, specifically one using an evil twin of a Wi-Fi AP. There were seven targeted devices, the original AP to which they were connected, an evil twin access point and attacking laptop.

The seven devices were a coffee maker, a crockpot, which is also known as a slow cooker, an electricity switch, a motion detector, a programmable relay and two smartphones based on Android and Windows Phone. They were all connected to the original AP at the beginning of each test. The target devices were tested one by one and moved approximately 20 m away from the original AP, closer to the evil twin. This was to ensure that the signal of the evil twin was stronger than the original's. The evil twin was a regular AP with its SSID set to same as that of the original AP. The evil twin's setup page was monitored to see whether a targeted device had connected to the evil twin AP. The attacking laptop was used to drop devices off the original Wi-Fi. This design is visualized in Fig. 2. Deauthentication was done using the aireplay-ng command of the aircrack-ng program suite. The exact command to drop off the devices was

```
aireplay-ng -0 0 -a 00:0D:0B:67:83:CB -c 94:10:3E:5A:47:19 wlan1
```

The aim was to determine whether the devices would connect to an open Wi-Fi, or to a secured one that had the same security protocol as the original AP, in this case, WPA2. Different methods were tested to make a device switch from the original AP to the evil twin. The first technique was deauthentication. The attacking laptop continuously sent deauthentication messages to the original Wi-Fi network,

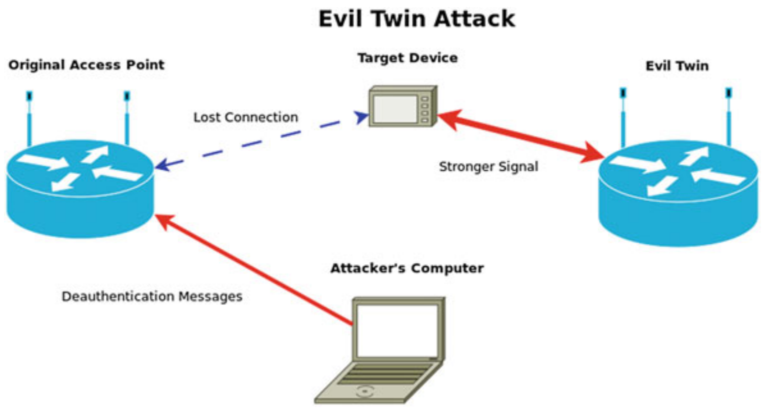

Fig. 2 The design of an Evil Twin attack. Target device is dropped out of the original access point's network with deauthentication messages. Then, the target is likely to connect to the access point's evil twin

disconnecting the device from the network and preventing reconnection. Next, the original AP was shut down. The last test was to keep the original Wi-Fi down and momentarily cut the power from the target.

3.4 Results

The devices, appliances and phones, were tested for two evil twin types: an open one and a secure one. These are marked as the rows of Table 6. The columns signify the efforts needed to make a device connect to the evil twin, and the last one is for devices that wouldn't connect to said network at all. Devices are listed by the easiest category in which they connected to the evil twin. Setting up an open network is easier for an attacker than figuring out the password of the original AP. So, if a device connects to an open evil twin, the device is listed on that row in the table. The efforts listed in columns grow more difficult to accomplish as they get to the right. Deauthentication is the easiest method to use, and turning the original AP off and rebooting the target device is the hardest. So, the coffee maker is the weakest link, needing only deauthentication to enter an open Wi-Fi. It isn't listed in any other cell, as it already connected in an easier scenario. The most secure devices, the electricity switch, the motion detector and the Windows phone, are in the lower right, as they didn't connect in any situation.

First, the evil twin was set up as an open Wi-Fi that had the same name as the original network. No device connected to the evil twin at this point. Then, the laptop sent deauthentication messages to the original Wi-Fi in order to disconnect the targeted device from the original AP and prevent it from reconnecting. Unable to form a connection to the original AP, the coffee maker connected to the evil twin.

Table 6 Successful evil twin attacks

	Open Wi-Fi and deauthentication	Open Wi-Fi and orig. AP off and device rebooted	WPA2 and deauthentication	WPA2 and original AP off	WPA2 and original AP off and target device rebooted
Coffee maker	X	X	X	X	X
Crockpot	–	–	X	X	X
Android phone	–	–	–	X	X
Electricity switch	–	–	–	–	–
Motion detector	–	–	–	–	–
Programmable relay	–	–	–	X	X
Android phone	–	–	–	–	–
Windows phone	–	–	–	–	–

This is the most worrying scenario, as the attacker doesn't need the original Wi-Fi's password, he or she only needs to know the SSID. Connecting to a similarly named open network is a dangerous solution, as open networks are inherently insecure. One comforting result was that other devices didn't connect to the open Wi-Fi. Even shutting down the original AP and rebooting the devices didn't make the more secure devices connect to an unsafe network.

Next, the evil twin used the same security protocol, WPA2, that the original AP used. The Wi-Fi password was the same in both networks. A deauthentication attack made the crockpot switch to the evil twin, and shutting down the original AP altogether caused the programmable relay to connect to the evil twin. The switch, the motion detector, and the smartphones didn't connect to the evil twin in any tested situation.

4 Discussion and Conclusions

Though small, cheap and not very powerful as single devices, smart home appliances, as well as all IoT devices, can cause great harm if they are out of control. Damages can be money loss, damage to property, damage to lives or disabled information systems. A crockpot could induce monetary loss if it was hacked. We have shown that it is possible for an attacker to gain access to a poorly protected network or even force the devices onto other networks and then take full or partial control of the devices. Thus, a crockpot could be turned on when no one is at home or during the night. Over time, the owner would pay the price for the electricity spent for nothing.

Property and even human lives would be endangered if said crockpot were to be set to heat up as hot as possible and something easily flammable was too near.

The risk induced for information systems is often overlooked, but compromised smart appliances can be used as entry points for going further into the protected network. From within the network, other vulnerable devices could easily be found and compromised. To automate this process and spread the infection within the network, a worm could be created. Detecting such a worm would be rather difficult, as varying ports, timings and even protocols could be used to distribute the worm. To counter this, strict controls and detection mechanisms should be implemented within the Wi-Fi network.

The breached devices can also harm systems outside their Wi-Fi network. As of late, there have been some massive DDoS attacks performed with huge botnets consisting of home routers, security cameras, digital video recorders, and other IoT devices. The botnets have generated traffic reaching close to a terabyte per second (Woolf 2016). The first analysis has already revealed that this has been made possible by the liberal default settings on the devices and the lack of awareness of the end users (Newman 2016; Caltum and Segal 2016). The number of affected devices in the wild has already reached millions. Fixing either of the causes, the liberal and widely known default settings of the devices or the security awareness of the end users, would very much improve the situation.

As shown, smart home devices have known vulnerabilities and cause a threat to physical surroundings and services on the Internet. Some simple remedies can, however, be found. Updating software and good passwords go a long way.

To use UPnP messages to turn on a coffee maker, the attacker needs to be in the same local network. This is easy if the network is wireless and doesn't use secure protocols; the attacker just logs in. If the Wi-Fi is secured, the attacker could try using an open evil twin. This way, there is no need to know the password or other authentication methods. At this point, the examined coffee maker would be in the attacker's control.

Brute forcing the AP's password is one way to try and enter a Wi-Fi. A good password is the precaution one can take. Depending on the attacker's motives, the password could be used to perform an MitM attack with the crockpot connected to this kind of evil twin. Physical access gives the attacker a new attack vector. Shutting down the original AP got two devices to switch to the fake Wi-Fi.

4.1 Future Work

Vulnerability of the Wi-Fi network enables the attacks discussed in this paper and many others. Whether this knowledge is taken to heart by the end users could be looked into by surveying how many homes or small offices use WPA2 or other encryption systems. Wardriving could collect valuable data of these usage rates.

In the light of the recent large botnet attacks, it would be interesting to search for possibilities to detect if your device is breached and has been used in DDoS attacks.

Securing devices could be looked into on a larger scale, such as how quickly manufacturers release updates that fix found vulnerabilities. Another direction for further research would be firmware analysis, possibly with automated or semiautomated tools such as Costin has used in his research (Costin 2015).

4.2 Remedies

The network for smart home devices can and should be protected as well as any other computer network. Basics like using encryption such as WPA2 in wireless networks and good passwords make any network a harder target for an attacker to gain access to. Also, the network could be further secured by utilizing the latest encryption mechanisms available and other restrictive methods, such as device MAC filtering, network segmentation, and firewalls. Also, intrusion detection systems (IDS) and intrusion prevention systems (IPS) should be implemented, if not at every users' home, then at least by the operator, to alarm the user or to monitor anomalous behavior within the network.

Updating the devices is crucial. If the studied devices were up to date, an attacker would not gain access to them, even with access to the network. Many smart home devices have updated firmware and software available, and the updates should be installed, even considering the risk of losing some functionality. In the long term, manufacturers should be paying more attention to updates and especially to the default settings of the devices. Unique default passwords and disabling remote access by default will provide a great increase in security, with some, rather small, discomfort to the users.

However, all the technical protection mechanisms implemented within the network can be circumvented if the attacker has physical access to the devices or the network infrastructure. Thus, this access must be limited to the persons necessary with physical limitations, including proper placement and locking mechanisms.

In summary, protect the network the devices are connected to, disable physical access to both the devices and the network, and keep the devices updated to protect your smart home.

References

Abomhara M, Køien GM (2014) Security and privacy in the Internet of Things: current status and open issues. In 2014 international conference on privacy and security in mobile systems (PRISMS), pp 1–8

Ashton K (2009) That 'Internet of Things'. RFID J http://www.rfidjournal.com/articles/view?4986. Accessed 18 Aug 2016

Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. Comput Netw 54(15):2787–2805

Baker WH, Hylender A, Pamula CD, Porter J, Spitler CM (2011) Data breach investigations report. Verizon RISK Team. https://www.researchgate.net/profile/Wade_Baker/publication/265027624_2011_Data_Breach_Investigations_Report_AUTHORS/links/5630d9ac08ae13bc6c35312c/2011-Data-Breach-Investigations-Report-AUTHORS.pdf, pp 1–72

Black Hat (2016) Let's See What's Out There—Mapping the Wireless IOT

Black Hat (2016) A Lightbulb Worm?

Caltum E, Segal O (2016) SSHowDowN: exploitation of IoT devices for launching mass-scale attack campaigns. https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/sshowdown-exploitation-of-iot-devices-for-launching-mass-scale-attack-campaigns.pdf. Accessed 14 Oct 2016

Command substitution. http://www.tldp.org/LDP/abs/html/commandsub.html. Accessed 26 Oct 2016

Conti M, Dragoni N, Lesyk V (2016) A survey of man in the middle attacks. IEEE Commun Surv Tutor 18(3), 2027–2051

Costin A (2015) Large Scale Security Analysis of Embedded Devices' Firmware. TELECOM ParisTech

Denning T, Kohno T, Levy HM (2013) Computer security and the modern home. Commun ACM 56(1):94–103

Dyn Statement on 10/21/2016 DDoS Attack | Dyn Blog. http://dyn.com/blog/dyn-statement-on-10212016-ddos-attack/. Accessed 26 Oct 2016

Eldaw E, Zeki AM, Senan S (2013) Analysis of wardriving activity and WiFi access points. In: Shaikh FK, Chowdhry BS, Ammari HM, Uqaili MA, Shah A (eds) Wireless sensor networks for developing countries. Springer, Heidelberg, pp 51–59

Ersue M, Romascanu D, Schoenwaelder J, Sehgal A (2015) Management of networks with constrained devices: use cases. RFC Editor, RFC7548, May 2015

Gartner Says 6.4 Billion Connected. Gartner, Inc. Newsroom. http://www.gartner.com/newsroom/id/3165317. Accessed 18 Aug 2016

Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gener Comput Syst 29(7):1645–1660

Hart B (2015) My SecTor Story: Root Shell on the Belkin WeMo Switch," The State of Security, 25-Nov-2015. http://www.tripwire.com/state-of-security/featured/my-sector-story-root-shell-on-the-belkin-wemo-switch/. Accessed 24 Aug 2016

International Telecommunication Union, "X.1205: Overview of cybersecurity". ITU, April 2008

ITU internet reports 2005: The Internet of Things. http://www.itu.int/pub/S-POL-IR.IT-2005/e. Accessed 21 Sep 2016

Kim J, Lee J, Kim J, Yun J (2014) M2 M service platforms: survey, issues, and enabling technologies. IEEE Commun Surv Tutor 16(1):61–76

KrebsOnSecurity Hit With Record DDoS—KrebsonSecurity 2016

Kumar A, Paul P (2016) Security analysis and implementation of a simple method for prevention and detection against Evil Twin attack in IEEE 802.11 wireless LAN. In 2016 international conference on computational techniques in information and communication technologies (ICCTICT), pp 176–181

Kyaw AK, Tian Z, Cusack B (2016) Wi-Pi: a study of WLAN security in Auckland City. Int J Comput Sci Netw Secur IJCSNS 16(8):68–80

Lanze F, Panchenko A, Ponce-Alcaide I, Engel T (2015) Hacker's toolbox: detecting software-based 802.11 evil twin access points. In 2015 12th annual IEEE consumer communications and networking conference (CCNC), pp 225–232

Madakam S, Ramaswamy R, Tripathi S (2015) Internet of Things (IoT): a literature review. J Comput Commun 03(05):164–173

Mattern F, Floerkemeier C (2010) From the internet of computers to the internet of things. In From active data management to event-based systems and more. Springer, pp 242–259

Mineraud J, Mazhelis O, Su X, Tarkoma S (2016) A gap analysis of Internet-of-Things platforms. Comput Commun

Mustafa H, Xu W (2014) CETAD: detecting evil twin access point attacks in wireless hotspots. In 2014 IEEE conference on communications and network security (CNS), pp 238–246

Newman LH (2016) Akamai finds longtime security flaw in 2 million devices, WIRED. https://www.wired.com/2016/10/akamai-finds-longtime-security-flaw-2-million-devices/. Accessed 14 Oct 2016

Prowell S, Kraus R, Borkin M (2010) Seven deadliest network attacks. Elsevier

Radack S, Kuhn R (2012) Protecting wireless local area networks. IT Prof 14(6):59–61

Sarma S, Brock DL, Ashton K (2000) The networked physical world. Auto-ID Cent White Pap MIT-AUTOID-WH-001

Sheng Y, Tan K, Chen G, Kotz D, Campbell A (2008) Detecting 802.11 MAC layer spoofing using received signal strength. In The 27th conference on computer communications IEEE INFOCOM 2008

Song Y, Yang C, Gu G (2010) Who is peeping at your passwords at Starbucks?—To catch an evil twin access point. In 2010 IEEE/IFIP international conference on dependable systems networks (DSN), pp 323–332

Woolf N (2016) DDoS attack that disrupted internet was largest of its kind in history, experts say, The Guardian